ARTICLE IN PRESS

Statistics and Probability Letters ▮ (▮▮▮▮) ▮▮▮–▮▮▮

Contents lists available at ScienceDirect

Statistics and Probability Letters

journal homepage: www.elsevier.com/locate/stapro

A very fast algorithm for matrix factorization

Vladimir Nikulin [a], Tian-Hsiang Huang [b], Shu-Kay Ng [c], Suren I. Rathnayake [a,d], Geoffrey J. McLachlan [a,d,*]

[a] Department of Mathematics, University of Queensland, St Lucia, QLD, Australia
[b] Institute of Information Management, National Cheng Kung University, Tainan, Taiwan
[c] School of Medicine, Griffith University, Meadowbrook, QLD, Australia
[d] Institute for Molecular Bioscience, University of Queensland, St Lucia, QLD, Australia

ARTICLE INFO

Article history:
Available online xxxx

Keywords:
Matrix factorization
Nonnegative matrix factorization
High-dimensional data
Microarray gene-expression data
Supervised classification

ABSTRACT

We present a very fast algorithm for general matrix factorization of a data matrix for use in the statistical analysis of high-dimensional data via latent factors. Such data are prevalent across many application areas and generate an ever-increasing demand for methods of dimension reduction in order to undertake the statistical analysis of interest. Our algorithm uses a gradient-based approach which can be used with an arbitrary loss function provided the latter is differentiable. The speed and effectiveness of our algorithm for dimension reduction is demonstrated in the context of supervised classification of some real high-dimensional data sets from the bioinformatics literature.

1. Introduction

We let $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$ denote n observed p-dimensional observations, where the number of variables p is very large relative to n. For example, in the analysis of microarray gene-expression data, n (the number of tissues) might be only 50, whereas p (the number of genes) might be in the tens of thousands. We follow the traditional biologists' practice of letting

$$\boldsymbol{X} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)$$

be the $p \times n$ data matrix. The usual statistical practice is to take the transpose of $\boldsymbol{X}$, $\boldsymbol{X}^T$, as the data matrix. Without loss of generality, we assume that the overall mean of $\boldsymbol{X}$ is zero.

In most statistical analyses of the data matrix $\boldsymbol{X}$, some form of dimension reduction is required, typically before the primary analysis is performed, or with some approaches it might be done in conjunction with the main analysis. In recent times, much attention has been given to matrix factorizations of the form,

$$\boldsymbol{X} = \boldsymbol{AB}, \tag{1}$$

where $\boldsymbol{A}$ is a $p \times q$ matrix and $\boldsymbol{B}$ is a $q \times n$ matrix and where q is chosen to be much smaller than p. For a specified value of q, the matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ are chosen to minimize

$$\|\boldsymbol{X} - \boldsymbol{AB}\|^2, \tag{2}$$

where $\|\cdot\|$ is the Frobenius norm (the sum of squared elements of the matrix). With this factorization, dimension reduction is effected by replacing the data matrix $\boldsymbol{X}$ by the solution $\hat{\boldsymbol{B}}$ for the factor matrix $\boldsymbol{B}$; the ith row of $\hat{\boldsymbol{B}}$ gives the values of

* Corresponding author at: Department of Mathematics, University of Queensland, St Lucia, QLD, Australia.
E-mail addresses: gjm@maths.uq.edu.au, g.mclachlan@uq.edu.au (G.J. McLachlan).

0167-7152/$ – see front matter © 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.spl.2011.02.001

Please cite this article in press as: Nikulin, V., et al., A very fast algorithm for matrix factorization. Statistics and Probability Letters (2011), doi:10.1016/j.spl.2011.02.001

the ith metavariable for the n entities. Thus the original p variables are replaced by q metavariables. When the elements of $\boldsymbol{X}$ are nonnegative, we can restrict the elements of $\boldsymbol{A}$ and $\boldsymbol{B}$ to be nonnegative. This approach is called nonnegative matrix factorization (NMF) in the literature (Lee and Seung, 1999). We shall call the general approach where there are no constraints on $\boldsymbol{A}$ and $\boldsymbol{B}$, GMF (general matrix factorization).

Principal component studies may often provide a suitable first step in the analysis of complex data, prior to a subsequent and more in depth rigorous modelling of data (Mertens, 2003). The classic method for factoring the data matrix $\boldsymbol{X}$ is singular-value decomposition (SVD); see Golub and van Loan (1983). It follows from this theorem that we can decompose $\boldsymbol{X}$ exactly into the form

$$\boldsymbol{X} = \boldsymbol{L}\boldsymbol{D}\boldsymbol{R}^T, \tag{3}$$

where $\boldsymbol{L} = (\boldsymbol{l}_1, \ldots, \boldsymbol{l}_k)$ is a $p \times k$ matrix with orthonormal columns, $\boldsymbol{R} = (\boldsymbol{r}_1, \ldots, \boldsymbol{r}_k)$ is a $n \times k$ matrix with orthonormal columns, $\boldsymbol{D}$ is a diagonal matrix with elements $d_1 \geq d_2 \geq \cdots \geq d_k > 0$, and $k \leq \min(p, n)$ is the rank of $\boldsymbol{X}$. For any $q \leq k$,

$$\sum_{i=1}^{q} d_i \boldsymbol{l}_i \boldsymbol{r}_i^T = \arg \min_{\hat{\boldsymbol{X}} \in M(q)} \|\boldsymbol{X} - \hat{\boldsymbol{X}}\|^2, \tag{4}$$

where $M(q)$ is the set of rank-q $p \times n$ matrices; see, for example, Eckart and Young (1936).

Let $\boldsymbol{L}^{(q)} = (\boldsymbol{l}_1, \ldots, \boldsymbol{l}_q)$, $\boldsymbol{R}^{(q)} = (\boldsymbol{r}_1, \ldots, \boldsymbol{r}_q)$, and $\boldsymbol{D}^{(q)}$ be the diagonal matrix with diagonal elements $d_1, \ldots, d_q$. Then on considering the matrix factorization (1) of $\boldsymbol{X}$, it follows from (4) that for a specified value of q we can find the factor matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ that minimize (2) by taking $\hat{\boldsymbol{A}} = \boldsymbol{L}^{(q)}$ and $\boldsymbol{B} = \boldsymbol{D}^{(q)}\boldsymbol{R}^{(q)^T}$.

Now SVD, effectively principal component analysis (PCA), imposes orthogonality constraints on the rows of the matrix $\boldsymbol{B}$. However, this ignores the non-independence of biological processes, which is equivalent to nonorthogonality of the rows of $\boldsymbol{B}$; see, for example, Kossenkov and Ochs (2009). On the other hand, GMF which has no constraints on $\boldsymbol{B}$ provides a factorization into a lower-dimensional subspace with no orthogonality constraints on its basis vectors. Thus GMF has the flexibility to model, for example, biological behaviour in which the gene signatures overlap. In contrast, PCA with its orthogonality constraints is overly constraining for such data and is thus not suited to isolating gene signatures that have appreciable overlap. Also, PCA is based on finding the directions of greatest variance, but the sample covariance matrix provides misleading estimates where the number of variables p is much greater than the number n of observations (Johnstone and Lu, 2009). It should be noted too that the GMF approach is suitable for other high-throughput data where the fundamental assumptions of multiple overlapping sets within the data and nonorthogonality of these sets across observations hold.

Hence we consider a very fast approach to the general matrix factorization (1), using a gradient-based algorithm that is applicable to an arbitrary (differentiable) loss function. The novelty of our algorithm lies in the way that on each global iteration it

(a) iterates on only a small subset of the elements of the factor matrix $\boldsymbol{A}$ with the other factor matrix $\boldsymbol{B}$ fixed before reversing their roles;
(b) loops through all the terms in the objective function, minimizing them individually at a time rather than their total sum (that is, it adopts a stochastic gradient descent approach).

As to be presented in Section 3, our algorithm takes only between 10 and 15 s in performing 300 global iterations to provide a $q = 11$ rank factorization of a 2000×62 data matrix for the colon cancer data set of Alon et al. (1999). It is understandably much quicker than nonnegative matrix factorization (NMF) which requires the estimates of the individual elements of $\boldsymbol{A}$ and $\boldsymbol{B}$ to be nonnegative.

The effectiveness of our algorithm is to be demonstrated in its application to provide a reduction in the number of genes for use in the formation of classifiers in the supervised classification of five well-known high-dimensional data sets in the bioinformatics literature.

2. Background

Here we consider the factorization of $\boldsymbol{X}$ into $\boldsymbol{A}\boldsymbol{B}$ in the spirit that it has no real importance in and of itself other than as a computationally convenient means for obtaining a reduction in the number of variables. Of course in some situations in practice once the factorization has been made, attention will turn to the interpretability of the metavariables.

The latter consideration has led to much recent interest in the use of NMF in the analysis of data for which the elements are nonnegative. It constrains the elements of the factor matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ to be nonnegative, which can be advantageous from the point of view of interpretability. Lee and Seung (1999) and Lee and Seung (2001) developed NMF in order to improve upon the interpretability of SVD. The nonnegativity constraints on $\boldsymbol{A}$ and $\boldsymbol{B}$ form a whole in a nonsubtractive way. In this way, NMF is considered as a procedure for learning a parts-based representation (Lee and Seung, 1999). However, as pointed out in Li et al. (2001) the additive parts by NMF are not necessarily localized. This led them to propose a subspace method, called local nonnegative matrix factorization (LNMF) for learning spatially localized, parts-based representation of visual patterns; see also Donoho and Stodden (2004), Gao and Church (2005), Fogel et al. (2007), and the recent monograph Cichocki et al. (2010).

Please cite this article in press as: Nikulin, V., et al., A very fast algorithm for matrix factorization. Statistics and Probability Letters (2011), doi:10.1016/j.spl.2011.02.001

More recently, Ding et al. (2010) has considered variations of NMF where the elements of $\boldsymbol{A}$, but not of $\boldsymbol{B}$, are constrained to be nonnegative, and so allowing the data matrix $\boldsymbol{X}$ to have mixed signs (semi-NMF). They also consider algorithms in which the basis vectors of $\boldsymbol{A}$ are constrained to be convex combinations of the data points. In other work, Witten et al. (2009) have proposed a penalized matrix decomposition for computing a q-rank approximation to $\boldsymbol{X}$.

3. Gradient-based algorithm for GMF

We now describe our gradient-based algorithm for carrying out the general matrix factorization (1) of the data matrix $\boldsymbol{X}$. In this paper we apply this algorithm using always a squared-error loss function. However, its use is not restricted to the latter loss function, and so it can be used to minimize the objective function given by

$$L(\boldsymbol{A}, \boldsymbol{B}) = \frac{1}{p \cdot n} \sum_{i=1}^{p} \sum_{j=1}^{n} \Psi(E_{ij}), \tag{5}$$

where $E_{ij} = x_{ij} - \sum_{f=1}^{q} a_{if} b_{fj}$, and Ψ is the loss function assumed to be differentiable with derivative denoted by ψ.

For illustrative purposes here in the description of the algorithm, we take Ψ to be a member of the exponential family of loss functions given by

$$\Psi(x; \alpha) = 2\frac{(\cosh(\alpha x) - 1)}{\alpha^2} = \alpha^{-2}(\exp(\alpha x) + \exp(-\alpha x) - 2), \tag{6}$$

where α is a regularization parameter. Note that a squared loss function may be regarded as a marginal limit in relation to this family of loss functions since

$$\lim_{\alpha \to 0} \Psi(x; \alpha) = x^2. \tag{7}$$

In our initial experiments Nikulin and McLachlan (2009), we tried a range of values between 0.003 and 0.004 for α, which gave similar results as for the squared loss function.

The algorithm can be implemented as follows.

Gradient-based framework for matrix factorization

1: Input: $\boldsymbol{X}$-matrix of microarrays.
2: Select m-number of global iterations; q-number of factors; $\lambda > 0$-initial learning rate, $0 < \xi < 1$-correction rate, L_S-initial value of the target function.
3: Initial matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ may be generated randomly.
4: Global cycle: repeat m times the following steps 5–17:
5: genes-cycle: for $i = 1$ to p repeat steps 6–15:
6: tissues-cycle: for $j = 1$ to n repeat steps 7–15:
7: compute prediction $S = \sum_{f=1}^{q} a_{if} b_{fj}$;
8: compute error of prediction: $E = x_{ij} - S$;
9: internal factors-cycle: for $f = 1$ to q repeat steps 10–15:
10: compute $\alpha = a_{if} b_{fj}$;
11: update $a_{if} \Leftarrow a_{if} + \lambda \psi(E) b_{fj}$;
12: $E \Leftarrow E + \alpha - a_{if} b_{fj}$;
13: compute $\alpha = a_{if} b_{fj}$;
14: update $b_{fj} \Leftarrow b_{fj} + \lambda \psi(E) a_{if}$;
15: $E \Leftarrow E + \alpha - a_{if} b_{fj}$;
16: compute $L = L(\boldsymbol{A}, \boldsymbol{B})$;
17: $L_S = L$ if $L < L_S$; otherwise: $\lambda \Leftarrow \lambda \cdot \xi$.
18: Output: $\boldsymbol{A}$ and $\boldsymbol{B}$-matrices of loadings and metagenes.

The following partial derivatives are necessary for the above algorithm (see steps 11 and 14):

$$\frac{\partial \Psi(E_{ij})}{\partial a_{if}} = -\psi(E_{ij}) b_{fj}, \tag{8}$$

$$\frac{\partial \Psi(E_{ij})}{\partial b_{fj}} = -\psi(E_{ij}) a_{if}. \tag{9}$$

The target function (5) that needs to be minimized includes a total of $q(p + n)$ regularization parameters. The algorithm loops through all the differences E_{ij}, minimizing them as a function of the elements of the two factor matrices $\boldsymbol{A}$ and $\boldsymbol{B}$. If the optimization were to be performed by fixing on $\boldsymbol{B}$ and solving the optimization with respect to $\boldsymbol{A}$ and then reversing the roles of the variables with the intention to iterate until convergence, there can be difficulties with convergence given that the two factor matrices are completely unconstrained. We circumvent this problem by iterating on only some of the elements of $\boldsymbol{A}$ before iterating on some of the elements of $\boldsymbol{B}$. This partial updating of $\boldsymbol{A}$ before a switch to a partial updating of $\boldsymbol{B}$ is very effective and is responsible for the very fast convergence of the process.

Please cite this article in press as: Nikulin, V., et al., A very fast algorithm for matrix factorization. Statistics and Probability Letters (2011), doi:10.1016/j.spl.2011.02.001

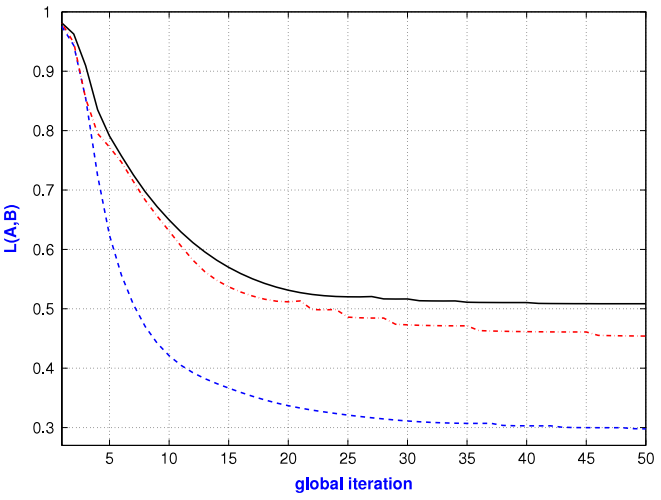

Fig. 1. Behaviour of the target (5) with squared loss as a function of global iteration for $q = 10$ metagenes; dashed blue, solid black and dot–dashed red lines correspond to the colon, leukaemia and lymphoma cases. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

This procedure of role reversal between the elements of the two factor matrices after only partial updating of their elements has been used effectively in the context of a recommender system; see, for example Paterek (2007) and, more recently, Koren (2009), who used factorization techniques to predict users' preferences for movies on the Netflix Prize data set.

It is noted that in the case of the squared loss function, we can optimize the value of the step-size. However, taking into account the complexity of the model, we recommend maintaining fixed and small values of the step-size or learning rate. In all our experiments we applied our algorithm using 100 global iterations with the following regulation parameters. The initial learning rate λ was set at 0.01, while the correction rate ξ rate was set at 0.75. The convergence of the algorithm is illustrated in Fig. 1 for GMF applied to the 2000×62 data matrix $\boldsymbol{X}$ for three data sets, including the colon cancer data set of Alon et al. (1999) with $n = 62$ tissues and $p = 2000$ genes. The other cancer data sets (leukaemia and lymphoma) are to be described in the next section. As pointed out in the introductory section, our algorithm takes only between 10 and 15 s in performing 300 global iterations to provide a $q = 11$ rank factorization of this data matrix compared to around 25 min to perform 20 global iterations with nonnegative matrix factorization. We used a Linux computer with speed 3.2 GHz, RAM 16 GB with the algorithm written in C.

4. Application of GMF in supervised classification

In what follows, we focus on the performance of GMF in its application to some data sets in the context of supervised classification (discriminant analysis). In this latter context, we have an obvious criterion to guide in the choice of the number q of metavariables, namely the estimated error rate of the classifier.

Concerning suitable estimates for the error rate of a classifier, we introduce the following notation. It is assumed that the observed data points $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$ come from g possible classes, $C_1, \ldots, C_g$, with known class labels specified by $\boldsymbol{z}$, where

$$\boldsymbol{z} = (\boldsymbol{z}_1, \ldots, \boldsymbol{z}_n)^T,$$

and where $\boldsymbol{z}_j$ is a g-dimensional vector of zeros or ones with its ith element, z_{ij}, defined to be one if $\boldsymbol{x}_j$ comes from class C_i, and zero otherwise ($i = 1, \ldots, g;\ j = 1, \ldots, n$). For the allocation of an observation $\boldsymbol{x}_o$ to one of the g possible classes, we let $r(\boldsymbol{x}_o; \boldsymbol{X}, \boldsymbol{z})$ be a classifier formed from the training data $\boldsymbol{X}$ with its known class labels in $\boldsymbol{z}$, where $r(\boldsymbol{x}_o; \boldsymbol{X}, \boldsymbol{z})$ equal to i implies that $\boldsymbol{x}_o$ is assigned to class C_i ($i = 1, \ldots, g$). We shall henceforth abbreviate $r(\boldsymbol{x}_o; \boldsymbol{X}, \boldsymbol{z})$ to $r(\boldsymbol{x}_o; \boldsymbol{X})$. Also, we let $e(\boldsymbol{X})$ denote an estimate of the error rate of $r(\boldsymbol{x}_o; \boldsymbol{X}, \boldsymbol{z})$, where dependency of this estimate on $\boldsymbol{z}$ is also suppressed for brevity of expression. If we use, for example, n-fold cross-validation (that is, the leave-one-out estimate), then

$$e(\boldsymbol{X}) = n^{-1} \sum_{i=1}^{g} \sum_{j=1}^{n} z_{ij} H[i, r(\boldsymbol{x}_j; \boldsymbol{X}_{(j)})], \tag{10}$$

where the function $H[u, v]$ is defined to be equal to 1 if $u \neq v$, and zero otherwise and where $\boldsymbol{X}_{(j)}$ denotes $\boldsymbol{X}$ with $\boldsymbol{x}_j$ deleted. Finally, we let $\hat{\boldsymbol{B}}^{(q)}(\boldsymbol{X})$ denote the solution for $\boldsymbol{B}$ when GMF is applied to $\boldsymbol{X}$ for a specified value of q.

Please cite this article in press as: Nikulin, V., et al., A very fast algorithm for matrix factorization. Statistics and Probability Letters (2011), doi:10.1016/j.spl.2011.02.001

In the case where the full data matrix $\boldsymbol{X}$ is replaced by the reduced matrix $\hat{\boldsymbol{B}}^{(q)}(\boldsymbol{X})$ computed for a specified q, we can use (10) to estimate the expected error rate of the classifier formed from this reduced set. An estimate is given by

$$e_1(\hat{\boldsymbol{B}}^{(q)}(\boldsymbol{X})) = n^{-1}\sum_{i=1}^{g}\sum_{j=1}^{n} z_{ij}H[i, r(\boldsymbol{b}_j; \hat{\boldsymbol{B}}^{(q)}_{(j)}(\boldsymbol{X}))], \tag{11}$$

where $\boldsymbol{b}_j$ is the jth column of $\hat{\boldsymbol{B}}^{(q)}(\boldsymbol{X})$ and $\hat{\boldsymbol{B}}^{(q)}_{(j)}(\boldsymbol{X})$ denotes $\hat{\boldsymbol{B}}^{(q)}(\boldsymbol{X})$ with its jth column $\boldsymbol{b}_j$ deleted.

As pointed out by Ambroise and McLachlan (2002), this estimate will provide an optimistic assessment of the true error rate of the classifier, since the reduced data matrix $\hat{\boldsymbol{B}}^{(q)}(\boldsymbol{X})$ should be recomputed on each fold of the cross-validation; that is, in the right-hand side of (11), $\hat{\boldsymbol{B}}^{(q)}_{(j)}(\boldsymbol{X})$ should be replaced by $\hat{\boldsymbol{B}}^{(q)}(\boldsymbol{X}_{(j)})$, the reduced data matrix obtained by applying the GMF algorithm to $\boldsymbol{X}_{(j)}$, the data matrix $\boldsymbol{X}$ with its jth column deleted. This estimate can be written as

$$e_2(\hat{\boldsymbol{B}}^{(q)}(\boldsymbol{X})) = n^{-1}\sum_{i=1}^{g}\sum_{j=1}^{n} z_{ij}H[i, r(\boldsymbol{b}_j; \hat{\boldsymbol{B}}^{(q)}(\boldsymbol{X}_{(j)}))]. \tag{12}$$

In order to calculate this estimated error rate with n-fold cross-validation, it means that the GMF algorithm has to be run n times in addition to its replication to the full data set. This is feasible given the speed with which the algorithm carries out the GMF. It should be pointed out that since the GMF does not make use of the known class labels, the selection bias of the classifier based on the selected subset of metavariables $\hat{\boldsymbol{B}}^{(q)}$ will not be nearly as great in magnitude as with selection methods that use the class labels. Also, in practice, n-fold cross-validation can produce an estimate with too much variability and so five- or ten-fold cross-validation is often used in a variance versus bias tradeoff (Ambroise and McLachlan, 2002).

We can choose the final value of q by taking it to be the value q_o that minimizes the estimated error rate $e_2(\hat{\boldsymbol{B}}^{(q)}(\boldsymbol{X}))$;

$$q_o = \arg\min_{q\in Q} e_2(\hat{\boldsymbol{B}}^{(q)}(\boldsymbol{X}_{(j)})), \tag{13}$$

where Q denotes the set of values considered for q. However, there is still a selection bias if we use

$$e_2(\hat{\boldsymbol{B}}^{(q_o)}(\boldsymbol{X})) = n^{-1}\sum_{i=1}^{g}\sum_{j=1}^{n} z_{ij}H[i, r(\boldsymbol{b}_j; \hat{\boldsymbol{B}}^{(q_o)}(\boldsymbol{X}_{(j)}))], \tag{14}$$

to estimate the error rate of the classifier based on the reduced set with the smallest error rate over the values of q considered; see, for example, Wood et al. (2007) and Zhu et al. (2008). We can correct for this bias by using the estimate

$$e_3(\hat{\boldsymbol{B}}^{(q_o)}) = n^{-1}\sum_{i=1}^{g}\sum_{j=1}^{n} z_{ij}H[i, r(\boldsymbol{b}_j; \hat{\boldsymbol{B}}^{(q_{oj})}(\boldsymbol{X}_{(j)}))], \tag{15}$$

where

$$q_{oj} = \arg\min_{q\in Q}\sum_{i=1}^{g}\sum_{\substack{j'=1\\ j'\neq j}}^{n}\frac{z_{ij'}H[i, r(\boldsymbol{b}_{j'}; \boldsymbol{B}^{(q)}(\boldsymbol{X}_{(j,j')}))]}{n-1}, \tag{16}$$

and $\boldsymbol{X}_{(j,j')}$ denotes the data matrix $\boldsymbol{X}$ with $\boldsymbol{x}_j$ and $\boldsymbol{x}_{j'}$ deleted.

It can be seen from (16) that in order to calculate the cross-validated estimate (15), we need to perform the GMF $n(n-1)$ times in addition to the original application to the full data set $\boldsymbol{X}$. This is still feasible since GMF can be implemented so quickly, although the total computational time becomes large as n increases. As noted above, using, say, ten-fold cross-validation would reduce the number of times that GMF has to be employed. In the data sets considered here, the increase in the estimated error rate given by the use of $e_3(\hat{\boldsymbol{B}}^{(q_o)})$ over (14) was very small (not of practical significance).

5. Supervised classification of some cancer data sets

We shall demonstrate the application of the GMF for dimension reduction in the context of supervised classification of five cancer data sets that have been commonly analysed in the bioinformatics literature, as briefly described in the following section.

5.1. Five medical data sets

For the colon data set (Alon et al., 1999) the data matrix $\boldsymbol{X}$ contains the expression levels of $p = 2000$ genes in each of $n = 62$ tissue samples consisting of $n_1 = 40$ tumours and $n_2 = 22$ normals.

Please cite this article in press as: Nikulin, V., et al., A very fast algorithm for matrix factorization. Statistics and Probability Letters (2011), doi:10.1016/j.spl.2011.02.001

Table 1
Some selected experimental results, where numbers in brackets in the first column indicate numbers of classes in the corresponding data set, and numbers of misclassified entries in the fourth, fifth and sixth columns. Results in the sixth column "NSC" were obtained using nearest-shrunken centroid method with threshold parameter Δ as it was described in Tibshirani et al. (2002). The column p_s indicates the number of used/selected features.

Data	Model	q	e_1	e_2	NSC	p_s	Δ
Colon (2)	SVM	8	0.08 (5)	0.113 (7)	0.129 (8)	141	1.3
Leukaemia (2)	SVM	25	0 (0)	0.014 (1)	0.014 (1)	73	1.9
Lymphoma (3)	MLR	10	0.032 (2)	0.032 (2)	0.016 (1)	3336	0.8
Breast (2)	SVM	18	0.1 (6)	0.15 (9)	0.2 (12)	53	1.2
Khan (4)	MLR	21	0.024 (2)	0.048 (4)	0.06 (5)	464	1.8

The data matrix for the leukaemia data set (Golub et al., 1999) contains the expression levels of $p = 7129$ genes for each of $n = 72$ patients, consisting of $n_1 = 47$ patients suffering from acute lymphoblastic leukaemia (ALL) and $n_2 = 25$ patients suffering from acute myeloid leukaemia (AML).

We followed the pre-processing steps of (Golub et al., 1999) applied to the leukaemia set: (1) thresholding: floor of 1 and ceiling of 20 000; (2) filtering: exclusion of genes with max/min ≤ 2 and (max–min) ≤ 100, where max and min refer respectively to the maximum and minimum expression levels of a particular gene across the tissue samples. This left us with $p = 1896$ genes. In addition, the natural logarithm of the expression levels was taken.

The data matrix for the lymphoma data set (Alizadeh et al., 2000) contains the gene-expression levels of the three most prevalent adult lymphoid malignancies: $n_1 = 42$ samples of diffuse large B-cell lymphoma (DLCL), $n_2 = 9$ samples of follicular lymphoma (FL), and $n_3 = 11$ samples of chronic lymphocytic leukaemia (CLL). The total sample size is thus $n = 62$ and there are $p = 4026$ genes.

The Sharma data set was described in Sharma et al. (2005) and contains the expression levels (mRNA) of $p = 1368$ genes for each of 60 blood samples taken from 56 women. Each sample was labelled by clinicians, with $n_1 = 24$ labelled as having breast cancer and $n_2 = 36$ labelled as not having it. Some of the samples were analysed more than once in separate batches giving a total of $n = 102$ labelled samples.

The fifth data set (Khan et al., 2001) contains the expression levels of $p = 2308$ genes for each of $n = 83$ tissue samples, each from a child who was determined by clinicians to have a type of small round blue cell tumour. This includes the following $g = 4$ classes: neuroblastoma (N), rhabdomyosarcoma (R), Burkitt lymphoma (B) and the Ewing sarcoma (E). The numbers in each class are: N($n_1 = 18$), R($n_2 = 25$), B($n_3 = 11$), and E($n_4 = 29$).

We applied double normalization to each data set. Firstly, we normalized each column to have means zero and unit standard deviations. Then we applied the same normalization to each row.

5.2. RSCTC data

Six microarray data sets were obtained from the Rough Sets and Current Trends in Computing (RSCTC) Challenge which was held as an special event at the RSCTC 2010 Conference in Poland. A description of these data sets is given in Table 2, further information can be obtained from Wojnarski et al. (2010). During this experiment, we applied GMF to the RSCTC training data.

Due to the large amount of genes present in these data sets, we have followed the gene selection and ranking procedure described in EMMIX-GENE (McLachlan et al., 2002). The EMMIX-GENE software automatically assesses the relevance of each of the p genes by fitting one and two (in some cases three) component t-mixture models to the expression data over the N samples for each gene considered individually. The relevance of a gene for clustering the tissue samples is assessed on the basis of the value of $-2\log(\lambda)$, where λ is the likelihood ratio statistic for testing $g = g_0$ versus $g = g_0 + 1$ components. If $-2\log(\lambda)$ for $g = 1$ versus $g = 2$ is greater than a specified threshold b_1, then the gene is taken to be relevant provided that smaller cluster in the two component mixture model exceeds a specified threshold b_2. If the $-2\log(\lambda) > b_1$ condition holds but the cluster size condition does not, then a three component t-mixture model is fitted to the tissue samples on this gene, and $-2\log(\lambda)$ is calculated to test $g = 2$ versus $g = 3$. Then the gene is selected as being relevant only if the $-2\log(\lambda) > b_1$ and at least two clusters implied by the $g = 3$ solution have sizes not less than b_2.

Ranking of the genes selected by the EMMIX-GENE was done by the their final $-2\log(\lambda)$ value, where genes with higher $-2\log(\lambda)$ are ranked to be more relevant. The top p_0 genes from each of data set was used to obtain metagenes through applying GMF.

5.3. Error rates for classifiers formed on the basis of metagenes

In Fig. 2, we plot the cross-validated error rate e_1 versus the number of metagenes q for four of the five medical data sets, using the support vector machine (SVM) in the case of $g = 2$ classes and (multinomial) logistic regression (LR) in the case of $g > 2$ classes. We did not plot e_1 for the leukaemia data set as it was close to zero if $q \geq 10$.

In Table 1, we list the cross-validated error rates e_1 and its bias-corrected version e_2 for each of the four medical data sets, where the classifier (SVM or MLR) is formed on the basis of q metagenes. To provide a guide to the level of performance of these classifiers, we also list the value of the error rate using the nearest-shrunken centroid method (Tibshirani et al., 2002).

Please cite this article in press as: Nikulin, V., et al., A very fast algorithm for matrix factorization. Statistics and Probability Letters (2011), doi:10.1016/j.spl.2011.02.001

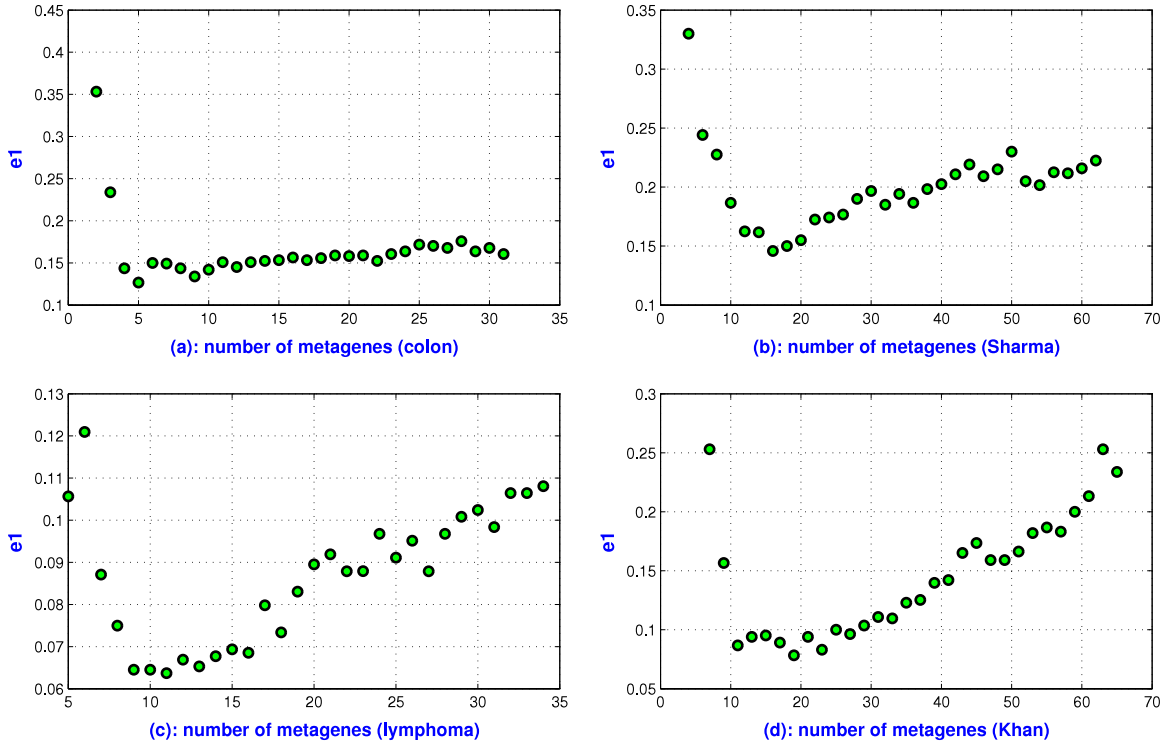

Fig. 2. The estimated error rate e_1 as a function of the number of metagenes.

Table 2
Description of the RSCTC Basic Track data: #-sequential index of the data set; N-numbers of samples; p-number of features/genes; n_1, n_2, n_3, n_4, n_5-number of samples per class in the training set.

#	N	p	K	n_1	n_2	n_3	n_4	n_5
1	123	54 675	2	88	35			
2	105	22 283	3	40	58	7		
3	95	22 277	5	23	5	27	20	20
4	113	54 675	5	11	31	51	10	10
5	89	54 613	4	16	10	43	20	
6	92	59 004	5	11	7	14	53	7

The bias-corrected error rate e_2 is smaller than that of the NSC method for all but one of the data sets (the lymphoma set). The estimated error rate for the nearest-shrunken method corresponds to e_2 in that it can be regarded as an almost unbiased estimate for a given subset of the genes, but it has not been corrected for bias over the set of values q considered; see Wood et al. (2007) and Zhu et al. (2008).

Considering now the six RSCTC data sets, the top p_0 genes from each of the RSCTC data sets were used to obtain $\hat{\boldsymbol{B}}$ with $q = 1, \ldots, 40$. GMF was applied to the top p_0 ranked genes of the genes selected by EMMIX-GENE. Table 3 illustrates estimates e_1 and e_2 for RSCTC data with GMF parameters $m = 100$, $\lambda = 0.01$, $\xi = 0.75$ and $L_s = 1$. In Table 4, slightly different GMF parameters were used with $m = 300$, $\lambda = 0.01$, $\xi = 0.85$ and $L_s = 1$, and linear SVM classifiers were used for all six data sets. When more than two classes were present in data, one-against-one voting scheme with $K(K-1)/2$ binary classifiers were used.

In some cases a preliminary opinion regarding the importance of the particular metavariables (metagenes) maybe derived after inspection of the heat maps. To illustrate this, we have plotted the heat maps in Fig. 3 for three of the data sets.

In this figure, we have sorted the tissues into their classes in order to consider visual differences between the patterns. In the case of the colon data in Fig. 3(a), we cannot see clear separation of the negative and positive classes. In contrast, in the case of the leukaemia data in Fig. 3(b), metagene N2 separates the first 47 tissues (from the top) from the remaining 25 tissues with only one exception. It is tissue 58, which is the only one misclassified tissue in Table 1 (cases $q = 3, 4$). Similarly, in the case of the lymphoma data in Fig. 3(c), metagene N1 separates clearly CLL from the remaining two classes. Further, metagene N3 separates DLCL from the remaining two classes.

To assist with the interpretation of the metagenes, we can examine the Gene Ontology (GO) (The GO Consortium, 2009) and the pathway records of the Kyoto Encyclopedia of Genes and Genomes (KEGG) (Kanehisa et al., 2010) for those genes that have high (absolute) correlations with the metagenes.

Please cite this article in press as: Nikulin, V., et al., A very fast algorithm for matrix factorization. Statistics and Probability Letters (2011), doi:10.1016/j.spl.2011.02.001

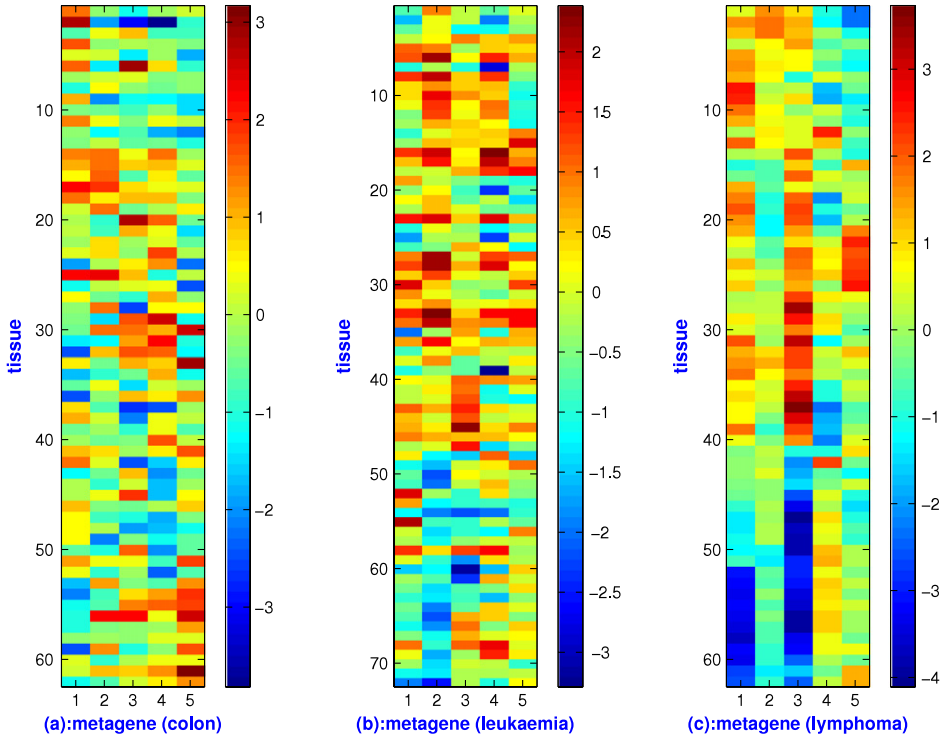

Fig. 3. Images of the matrix $\boldsymbol{B}$ for $q = 5$: (a) colon (sorted from the top: 40 positive then 22 negative), (b) leukaemia (sorted from the top: 47 ALL, then 25 AML) and (c) lymphoma (sorted from the top: 42 DLCL, then 9 FL, last 11 CLL). All three matrices were produced using the GMF algorithm with 100 global iterations as described in the text.

Table 3
The presented estimates e_1 and e_2 were calculated using RSCTC data; # refers to the sequential index of the data set. The number of misclassified samples are given in brackets.

#	Model	$p_0 = 1000$				$p_0 = 2000$			
		e_1	q	e_2	q	e_1	q	e_2	q
1	SVM	0.113 (14)	17	0.113 (14)	27	0.105 (13)	34	0.105 (13)	25
2	MLR	0.295 (31)	32	0.305 (30)	20	0.305 (32)	34	0.314 (33)	28
3	MLR	0.116 (11)	6	0.116 (11)	32	0.179 (17)	11	0.084 (8)	32
4	MLR	0.469 (53)	9	0.460 (52)	16	0.460 (52)	2	0.469 (53)	37
5	MLR	0.326 (29)	6	0.314 (28)	7	0.337 (30)	3	0.281 (25)	27
6	MLR	0.337 (31)	8	0.293 (27)	39	0.315 (29)	4	0.337 (27)	18

Table 4
The presented estimates e_1 and e_2 were calculated using RSCTC data; # refers to the sequential index of the data set. The number of misclassified samples are shown within brackets.

#	$p_0 = 1000$				$p_0 = 2000$			
	e_1	q	e_2	q	e_1	q	e_2	q
1	0.106 (13)	34	0.106 (13)	16	0.106 (13)	9	0.098 (12)	32
2	0.295 (31)	33	0.314 (33)	31	0.305 (32)	30	0.304 (32)	30
3	0.053 (5)	16	0.116 (11)	33	0.095 (6)	19	0.156 (15)	19
4	0.389 (44)	32	0.407 (46)	10	0.327 (37)	30	0.407 (46)	35
5	0.303 (27)	24	0.303 (27)	7	0.292 (26)	34	0.292 (26)	39
6	0.217 (20)	32	0.293 (24)	16	0.228 (21)	14	0.239 (22)	15

6. Conclusions

We have presented an algorithm for performing extremely fast general matrix factorization (GMF) of a high-dimensional data matrix $\boldsymbol{X}$. In practice some form of dimension reduction is invariably needed if standard or even nonstandard methods of statistical analysis are to be employed to gain meaningful insight from high-dimensional data matrices. The algorithm undertakes the factorization using gradient-based optimization for an arbitrary (differentiable) loss function. The $p \times n$ data matrix $\boldsymbol{X}$ is approximated by the product of two matrices, $\hat{\boldsymbol{A}}\hat{\boldsymbol{B}}$, where the $q \times n$ factor matrix $\hat{\boldsymbol{B}}$ can be used in place of $\boldsymbol{X}$ for

Please cite this article in press as: Nikulin, V., et al., A very fast algorithm for matrix factorization. Statistics and Probability Letters (2011), doi:10.1016/j.spl.2011.02.001

a specified value of q taken to be much smaller than p. The n columns of $\hat{\boldsymbol{B}}$ contain the values of the q metavariables for each of the n observed data vectors. The stability of the algorithm depends essentially on a properly selected learning rate, which must not be too big. We can provide additional functions so that the learning rate will be reduced or increased depending on the current performance.

To demonstrate the usefulness of the reduced data matrix $\hat{\boldsymbol{B}}$ in the context of supervised classification, we applied it to the data matrices from five cancer data matrices of microarray gene expressions that have been commonly analysed in the medical and scientific literature. The classification of the microarrays (tissue samples) containing these gene expressions are of known classification with respect to g classes, where g varies from 2 to 4. The results suggest that GMF as implemented by our algorithm is effective in providing a reduced data matrix for their subsequent use in forming a classifier that was taken to be either the SVM in the case of $g = 2$ classes or logistic regression for $g > 2$ classes.

Some main issues associated with the use of GMF are the choice of the number q of metavariables (latent factors), the interpretability of the metavariables, and the need for prefiltering of the variables before the factorization. These issues are beyond the scope of this paper.

But briefly on these issues here, the choice of the number of metavariables q in supervised classification can be based on the estimated error rate of the classifier formed from the specified number of metavariables. The situation is not as straightforward in the context of cluster analysis where there are no training data of known origin even if there were some *a priori* knowledge about the existence of some group structure in the data. One way to proceed in this context is to use the stability in the clustering as the level of q is varied as a guide to its final choice (Brunet et al., 2004; Tamayo et al., 2007). The question of whether there is a need for prefiltering in the context of cluster analysis has been considered recently by Zheng et al. (2009).

The problem of interpretability of the metavariables is generally not as straightforward as with NMF's since the latter are nonsubtractive combinations of the original variables (Zheng et al., 2009). In general, we can calculate the correlations between the original variables and the metavariables. For microarray data, we are currently designing a program that automatically attempts to establish links between genes highly correlated with a metagene and the Gene Ontology (The GO Consortium, 2009) and the pathway records of the Kyoto Encyclopedia of Genes and Genomes (Kanehisa et al., 2010).

Acknowledgements

This work was supported by a grant from the Australian Research Council. The research work of Tian-Hsiang Huang was partly supported by the NSC 98-2911-I-006-026 project from the National Science Council, Taiwan.

The computational resources used in this work were partly provided by the Queensland Cyber Infrastructure Foundation.

References

Alizadeh, A., Eisen, M.B., Davis, R.E., Ma, C., Lossos, I.S., Rosenwal, A., Boldrick, J.C., Sabet, H., Tran, T., Yu, X., et al., 2000. Distinct types of diffuse large B-cell lymphoma identified by gene expression profiling. Nature 403, 503–511.

Alon, U., Barkai, N., Notterman, D.A., Gish, K., Ybarra, S., Mack, D., Levine, A.J., 1999. Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. Proceedings of the National Academy of Sciences of the United States of America 96, 6745–6750.

Ambroise, C., McLachlan, G.J., 2002. Selection bias in gene expression on the basis of microarray gene expression data. Proceedings of the National Academy of Sciences of the United States of America 99, 6562–6566.

Brunet, J.-P., Tamayo, P., Golub, T.R., Mesirov, J.P., 2004. Metagene projection for cross-platform, cross-species characterization of global transcriptional states. Proceedings of the National Academy of Sciences of the United States of America 101, 4164–4169.

Cichocki, A., Zdunek, R., Phan, A.H., Amari, S.-I., 2010. Nonnegative Matrix and Tensor Factorizations. Wiley, Chichester.

Ding, C., Li, T., Jordan, M.I., 2010. Convex and semi-nonnegative matrix factorizations. IEEE Transactions on Pattern Analysis and Machine Intelligence 32, 45–55.

Donoho, D., Stodden, V., 2004. When does non-negative matrix factorization give a correct decomposition into parts?. In: Advances in Neural Information Processing Systems, vol. 16. MIT Press, Cambridge, MA.

Eckart, C., Young, G., 1936. The approximation of one matrix by another of low rank. Psychometrika 1, 211–218.

Fogel, P., Young, S.S., Hawkins, D.M., Ledirac, N., 2007. Inferential, robust non-negative matrix factorization analysis of microarray data. Bioinformatics 23, 44–49.

Gao, Y., Church, G., 2005. Improving molecular cancer class discovery through sparse non-negative matrix factorization. Bioinformatics 21, 3970–3975.

Golub, T.R., Slonim, D.K., Tamayo, P., Huard, C., Gassenbeck, M., Mesirov, J.P., Coller, H., Loh, M.L., Downing, J.R., Caligiuri, M.A., et al., 1999. Molecular classification of cancer: class discovery. Science 286, 531–537.

Golub, G., van Loan, C., 1983. Matrix Computations. Johns Hopkins University Press, Baltimore.

Johnstone, I.M., Lu, A.U., 2009. On consistency and sparsity for principal components analysis in high dimensions. Journal of the American Statistical Association 104, 682–693.

Kanehisa, M., Goto, S., Furumichi, M., Tanabe, M., Hirakawa, M., 2010. KEGG for representation and analysis of molecular networks involving diseases and drugs. Nucleic Acids Research 38, D355–D360.

Khan, J., Wei, J., Ringner, M., Saal, L., Ladanyi, M., Westermann, F., Berthold, F., Schwab, M., 2001. Classification and diagnostic prediction of cancers using gene expression profiling and artificial neural networks. Nature Medicine 7, 673–679.

Koren, Y., 2009. Collaborative filtering with temporal dynamics. In: Proceedings of the 15th ACM SIGKDD Conference on Knowledge Discovery and Data Mining. Paris. pp. 447–455.

Kossenkov, A.V., Ochs, M.F., 2009. Matrix factorization for recovery of biological processes from microarray data. In: Johnson, M.L., Brand, L. (Eds.), Methods in Enzymology, vol. 467. Academic Press, New York, pp. 59–77.

Lee, D.D., Seung, H.S., 1999. Learning the parts of objects by non-negative matrix factorization. Nature 401, 788–791.

Lee, D.D., Seung, H.S., 2001. Algorithms for non-negative matrix factorization. In: Advances in Neural Information Processing System, vol. 13. MIT Press.

Li, S.Z., Hou, X.W., Zhang, H.J., Cheng, Q.S., 2001. Learning spatially localized parts-based representation. In: Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition. Hawaii. vol. I. pp. 207–212.

McLachlan, G.J., Bean, R.W., Peel, D., 2002. A mixture model-based approach to the clustering of microarray expression data. Bioinformatics 18, 413–422.
Mertens, B., 2003. Microarrays, pattern recognition and exploratory data analysis. Statistics in Medicine 22 (11), 1879–1899.
Nikulin, V., McLachlan, G.J., 2009. On a general method for matrix factorization applied to supervised classification. In: Chen, J., et al. (Eds.), Proceedings of 2009 IEEE International Conference on Bioinformatics and Biomedicine Workshop. Washington, DC. IEEE, Los Alamitos, California, pp. 43–48.
Paterek, A., 2007. Improving regularized singular value decomposition for collaborative filtering. In: KDD Cup. ACM, San Jose, CA, pp. 39–42.
Sharma, P., Tibshirani, R., Skaane, P., Urdal, P., Berghagen, H., 2005. Early detection of breast cancer based on gene-expression patterns in peripheral blood cells. Breast Cancer Research 7, R634–R644.
Tamayo, P., Brunet, J.-P., Scanfeld, D., Ebert, B.L., Gillette, M.A., Roberts, C.W.M., Mesirov, J.P., 2007. Proceedings of the National Academy of Sciences of the United States of America 104, 5959–5964.
The GO Consortium, 2009. Gene ontolog [Online]. Available: http://www.geneontology.org.
Tibshirani, R., Hastie, T., Narasimhan, B., Chu, G., 2002. Diagnosis of multiple cancer types by shrunken centroids of gene expression. Proceedings of the National Academy of Sciences of the United States of America 99, 6567–6572.
Witten, D.M., Tibshirani, R., Hastie, R., 2009. A penalized matrix decomposition, with applications to sparse principal components and canonical correlation analysis. Biostatistics 10, 515–534.
Wojnarski, M., Janusz, A., Nyugen, H.S., Bazan, J., Luo, C.J., Chen, Z., Hu, F., Wang, G., Guan, L., Luo, H., Gao, J., Shen, Y., Nikulin, V., Huang, T.H., McLachlan, G.J., Bosnjak, M., Gamberger, G., 2010. RSCTC'2010 discovery challenge: mining DNA microarray data for medical diagnosis and treatment. In: 7th Int. Conf. Rough Sets Current Trends Computing. RSCTC 2010, Warsaw, Poland, 2010. In: Lect. Notes Artif. Int., vol. 6086. pp. 4–19.
Wood, I., Visscher, P., Mengersen, K., 2007. Classification based upon expression data: bias and precision of error rates. Bioinformatics 23, 1363–1370.
Zheng, C., Huang, D., Zhang, L., Kong, X., 2009. Tumor clustering using nonnegative matrix factorizations with gene selection. IEEE Transactions on Information Technology in Biomedicine 13, 599–607.
Zhu, J.X., McLachlan, G.J., Ben-Tovim, L., Wood, I., 2008. On selection biases with prediction rules formed from gene expression data. Journal of Statistical Planning and Inference 38, 374–386.

Please cite this article in press as: Nikulin, V., et al., A very fast algorithm for matrix factorization. Statistics and Probability Letters (2011), doi:10.1016/j.spl.2011.02.001